OpenBiblio System Administration

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Thanks for being an OpenBiblio system administrator. We're glad that you are helping your library in this way.

1 Managing your libraries

OpenBiblio 1.0 introduced the ability to manage multiple libraries with a single OpenBiblio installation. Each library is considered a separate "site," with its own name, collection, address, hours, closed dates, and other settings.

1.1 Adding a site

1. Go to Admin → Sites.
2. Click "Add New".
3. Fill out the relevant information. Choose the calendar that includes the dates your new library will be closed. See below for more information about the calendars.
4. Click "Add."

1.2 Deleting a site

1. Ensure that there are no copies at the site that you'd like to delete.
2. Sign in as a different site.
3. Go to Admin → Sites.
4. Delete the site.

Warning
Don't delete a site when you are logged in, or you will cause a permanent error state that won't be resolved until you manually add the site back into the database.

Warning
If you delete a site that has copies attached, they will show as "Undefined" in the system.

---

2 Managing your users

To add a new staff member, go to Admin → Staff Admin. Click on the "Add New" button, found both at the top and bottom of this screen. You can then assign a username, password, name, and permission set to your staff member.

Warning
You must assign at least one permission, or the staff member will not be able to log in.

2.1 Working with passwords

Openbiblio's password management system is very straightforward. Users with Admin permission are able to change their own password or the password of any other staff account (including other administrators). Users without the Admin permission may not change anybody's password, not even their own.

To change the password of a staff account, go to Admin → Staff Admin. Click the pwd button next to the account you are working with. Enter the new password.

3 Managing your catalog

This section describes the following settings in the Admin menu.

3.1 Managing media types

Media types link

3.2 Managing collections

Collections

3.3 Managing displayed fields

Biblio fields

3.4 Managing custom fields for copies

Biblio copy fields

3.5 Managing copy catalog options

Z39.50 and SRU

4 Configuring the OPAC

This section will describe settings found in the admin menu.

5 Managing plugins

OpenBiblio has a very flexible plugin system.

5.1 Creating a new plugin

This requires basic PHP skills and access to the files.

5.1.1 Design the plugin

1. Create a new directory that begins with the word plugin, followed by an underscore, followed by the name of your plugin. In UNIX systems, you might type something like this:

```
mkdir plugin_excitingPlugin
```

2. If you need to include custom CSS or JS, add it to a file called custom_head.php in this new directory.
3. Create three PHP files in the directory: a JS file, a Srvr file, and a Forms file.

Create a plugin server

Your plugin's server — located at `plugin_excitingPlugin/excitingSrvr.php` — is responsible for three tasks:

1. Performing any actions the user requests.
2. Obtaining any relevant data from the database, file structure, or external API.
3. Displaying the relevant data to the user in a nice HTML format.

A basic plugin server consists of a require statement to get the necessary data and methods and a switch statement to take care of the `$_REQUEST['mode']` variable. Here's a simple example:

```
<?php
        require_once('../shared/common.php');
        switch ($_REQUEST['mode']){
                case 'exciting':
                        echo 'This plugin is very exciting.';
                        break;
                case 'boring':
                        echo 'This plugin is pretty boring.';
                        break;
                default:
                        echo T('invalid mode');
                        break;
        }
```

If you would like to access data from the database, you should require the appropriate model from the model directory. For example, if you'd like to use the Biblios model, you would include this in your plugin_excitingPlugin/excitingSrvr.php:

```
require_once('../model/Biblios.php');
$bib = new Biblios;
```

You can then access all those great methods and data through the `$bib` object.

5.1.2 Add your plugin to the OpenBiblio menu

1. Create a new directory inside the plugins directory with the name of your plugin. In UNIX systems, you might type something like this:

```
mkdir plugins/excitingPlugin
```

2. Create a file within this new directory called nav.nav

```
<?php
Nav::node('tools/excitingPlugin', 'Exciting Plugin', '../plugin_excitingPlugin/ ↩
    excitingPluginForms.php');
```

3. If you'd like to use Obib's localization framework but need to localize some more terms, create a file called tran.tran. You can use this to add new terms to your locale.

> **Tip**
> Plugins already have all of the exciting localization ability built in. However, note that you cannot override localized term definitions within tran.tran.

Warning
There is not yet a place to specify localized terms for plugins.

+ . Turn on your plugin using Tools → Plugin Manager.

6 Managing locales

OpenBiblio can be translated into any language that you wish!

6.1 Creating a new locale

This requires access to the files and a good UTF-8 text editor.

6.1.1 Create a directory for your new locale

1. Create a new directory in the locale directory with the 2-character ISO 639-1 code for the language. For example, if you would like to do an Amharic translation, your directory would be called `am`.
2. Create a PHP file called metadata.php. Make sure that the class name begins with the ISO 639-1 code for your language.

```
<?php
class amMetaData {
        public function __construct() {
                $this->locale_description = "Amharic";
        }
        function pluralForm($n) {
                if ($n == 1 or $n == -1) {
                        return 'singular';
                } else {
                        return 'plural';
                }
        }
        function moneyFormat($amount) {
                if ($amount < 0) {
                        return sprintf("(-%.br)", abs($amount));
                } else {
                        return sprintf("%.br", $amount);
                }
        }
}
```

3. Add translations to trans.php.

7 Managing the MySQL database

This section will describe the database

8 Managing the server

This section will discuss other server-level considerations.

8.1 Logs

This section will mention logs of particular interest to OpenBiblio administrators.